Toward Evaluating Robustness of Reinforcement Learning with Adversarial Policy

Xiang Zheng
City University of Hong Kong
xzheng235-c@my.cityu.edu.hk

Xingjun Ma
Fudan University
xingjunma@fudan.edu.cn

Shengjie Wang
Tsinghua University
wangsj23@mails.tsinghua.edu.cn

Xinyu Wang
Tencent Inc.
rainewang@tencent.com

Chao Shen
Xi'an Jiaotong University
chaoshen@mail.xjtu.edu.cn

Cong Wang*
City University of Hong Kong
congwang@cityu.edu.hk

***Abstract*—Reinforcement learning agents are susceptible to evasion attacks during deployment. In single-agent environments, these attacks can occur through imperceptible perturbations injected into the inputs of the victim policy network. In multi-agent environments, an attacker can manipulate an adversarial opponent to influence the victim policy's observations indirectly. While adversarial policies offer a promising technique to craft such attacks, current methods are either sample-inefficient due to poor exploration strategies or require extra surrogate model training under the black-box assumption. To address these challenges, in this paper, we propose Intrinsically Motivated Adversarial Policy (IMAP) for efficient black-box adversarial policy learning in both single- and multi-agent environments. We formulate four types of adversarial intrinsic regularizers—maximizing the adversarial state coverage, policy coverage, risk, or divergence—to discover potential vulnerabilities of the victim policy in a principled way. We also present a novel bias-reduction method to balance the extrinsic objective and the adversarial intrinsic regularizers adaptively. Our experiments validate the effectiveness of the four types of adversarial intrinsic regularizers and the bias-reduction method in enhancing black-box adversarial policy learning across a variety of environments. Our IMAP successfully evades two types of defense methods, adversarial training and robust regularizer, decreasing the performance of the state-of-the-art robust WocaR-PPO agents by 34%-54% across four single-agent tasks. IMAP also achieves a state-of-the-art attacking success rate of 83.91% in the multi-agent game YouShallNotPass. Our code is available at https://github.com/x-zheng16/IMAP.**

***Index Terms*—Reinforcement learning, black-box evasion attack, adversarial policy, intrinsic motivation**

I. Introduction

A. Background

Reinforcement Learning (RL) agents are susceptible to a variety of attacks due to the vulnerabilities of their function approximators or policies themselves [1]. The growing application of RL agents in safety-critical systems, such as robotics and autonomous vehicles [2]–[5], underscores the need for the development of both certification methods [6]–[9] and empirical evaluation methods [10]–[13] to measure the robustness of deployed RL agents. Adversarial Policy (AP), a type of test-time evasion attack, has emerged as a crucial technique for assessing the robustness of the deployed RL engines or models [1], [11], [14]–[17].

In single-agent environments, AP is developed to generate imperceptible perturbations on the inputs of the victim policy network. Sun et al. [14] proposed generating action perturbation via AP first and then crafting the corresponding state perturbation via the Fast Gradient Sign Method (FGSM). Mo et al. [17] suggested using two APs to select the attack timing and determine the worst-case victim action separately. Apart from these white-box methods, Zhang et al. [1] introduced SA-RL to learn the optimal black-box AP in dense-reward locomotion tasks. However, SA-RL requires knowledge of the victim policy's training-time rewards, which are difficult for the adversary to obtain under the black-box threat model.

In multi-agent competitive environments, AP is used to control an opponent agent to interact with the victim agent and indirectly influence the observation of the victim. Gleave et al. [11] first discovered this type of AP in two-player zero-sum competitive games, denoted as AP-MARL. Wu et al. [15] suggested training an extra surrogate victim model by imitation learning first and then using an explainable Artificial Intelligent technique to identify the attack timing. Guo et al. [16] developed AP learning for non-zero-sum games by simultaneously maximizing the adversary's and minimizing the victim's value functions. However, training an additional surrogate victim model yields only a marginal improvement in the attacking success rate [15]. Moreover, all these AP learning methods are sample-inefficient due to their trivial dithering exploration methods.

B. Motivations and Design Rationale

a) Motivations: In this work, we explore and propose Intrinsically Motivated Adversarial Policy (IMAP) for efficient black-box AP learning in both single- and multi-agent environments. There are three main challenges. Firstly, efficient exploration is known to be critical for RL algorithms to improve performance and reduce sample complexity. However, existing AP learning methods all suffer from poor exploration in both single- and multi-agent environments as they all explore in an ad-hoc and trivial manner by heuristically perturbing the

*Corresponding author.

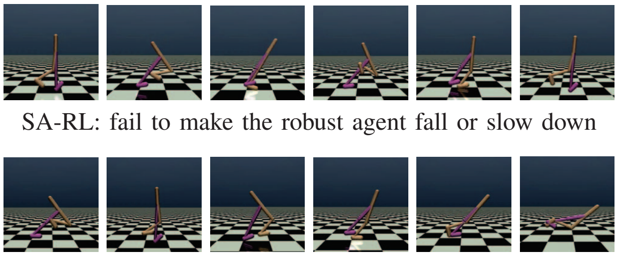
2158-3927/24/$31.00 ©2024 IEEE
DOI 10.1109/DSN58291.2024.00038

outputs of the AP with Gaussian noise. To address this, we design four types of adversarial intrinsic regularizers to enhance the exploration of the AP in a principled way. Adversarial intrinsic regularizers encourage the AP to explore novel states more efficiently so as to uncover potential vulnerabilities of the black-box victim policy. Secondly, the incorporation of adversarial intrinsic regularizers presents a new challenge: how to effectively balance the original extrinsic objective and the newly introduced adversarial intrinsic regularizers. To simplify the hyperparameter search for the optimal temperature parameter that controls the strength of the regularization, we employ constrained policy optimization to develop an adaptive balancing strategy. Thirdly, existing AP methods, except for AP-MARL, all follow a relaxed black-box threat model or require extra surrogate victim model training. One of the key assumptions on the knowledge of the adversary made by AP-MARL is that the adversary against the deployed victim policy does not have access to the training-time rewards and the value function of the deployed victim agent. To address this, we stick to the (unrelaxed) black-box assumptions on the knowledge of the adversary to design our IMAP and do not rely on extra surrogate victim models.

b) Design Rationale: To encourage the exploration of the AP, we design four types of adversarial intrinsic regularizers for IMAP that maximize the adversarial State Coverage (SC), Policy Coverage (PC), Risk (R), and Divergence (D). All four types of adversarial intrinsic regularizers are designed for the AP to uncover the potential vulnerabilities of the victim policy efficiently and have solid theoretical support, including state entropy [18], policy cover [19], constrained policy optimization [20], and policy diversity [21], [22]. Intuitively, efficient exploration for black-box AP learning can involve either uniform state visitation (maximizing the adversarial SC) or maximizing deviation from explored regions (maximizing the adversarial PC). Further, the R- and D-driven adversarial intrinsic regularizers are also well-motivated, with the former encouraging the AP to lure the victim policy into adversarial states and the latter encouraging the AP to keep deviating from its past policies to be diverse. In addition to promoting the exploration of the AP, the inductive bias introduced by adversarial intrinsic regularizers may distract the adversary from its objective—decreasing the performance of the victim policy—in the final stage of AP learning. We find that such a distraction phenomenon exists in sparse-reward tasks and design a novel bias-reduction method to enhance the performance of IMAP further.

Summary of Contributions. Our main contributions are summarized as follows:

- We propose IMAP—a general regularizer-based black-box AP learning method—and design four types of novel, well-motivated, and principled adversarial intrinsic regularizers, i.e., SC-, PC-, R-, and D-driven, in both single- and multi-agent environments.
- In single-agent environments, our IMAP outperforms the baseline SA-RL in four dense-reward locomotion tasks and nine sparse-reward tasks, including six locomotion, two navigation, and one manipulation tasks.



SA-RL: fail to make the robust agent fall or slow down

IMAP (ours): successfully make the robust agent fall

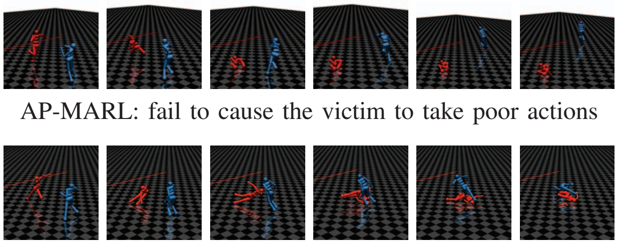
Fig. 1: The robust victim agent—trained with the state-of-the-art defense method WocaR [24]—is attacked by (**top**) the state-of-the-art AP method SA-RL and (**bottom**) our IMAP in the single-agent environment Walker. Though the WocaR Walker learned to lower its body to be robust, our IMAP can find its vulnerable states and successfully lure the victim to lean forward and fall.

- In single-agent environments, our IMAP successfully evades two types of state-of-the-art defense methods, including adversarial training (e.g., ATLA and ATLA-SA [1]) and robust regularizer (e.g., SA [8], RADIAL [23], and WocaR [24], shown in Fig. 1). We empirically show that a defense method that successfully defends one type of IMAP attack can fail against another type of IMAP, raising a new challenge for developing robust RL algorithms.
- In multi-agent environments, our IMAP achieves a state-of-the-art attacking success rate of 83.91% in the competitive game YouShallNotPass, outperforming the baseline AP-MARL [11]. The adversary learns a natural blocking skill with the policy-coverage-driven adversarial intrinsic regularizer, shown in Fig. 2. IMAP also outperforms AP-MARL in another competitive game, KickAndDefend.
- We develop a novel bias-reduction method for IMAP based on the adversarial optimality constraint and empirically demonstrate that it can effectively boost performance in both single- and multi-agent environments.

II. Preliminaries

We introduce the formulations of single- and multi-agent RL tasks and the basic policy optimization method in this section. In all tasks, the goal of the victim is to maximize its expected episode rewards, while the adversary aims to minimize the expected episode rewards of the victim.

a) Single-Agent RL Tasks: In single-agent tasks, the agent interacts with the environment by taking sequential actions according to the observed state at each step. This process is usually modeled as a Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, P, R_E, \gamma, \mu)$. $\mathcal{S}$ and $\mathcal{A}$ are the state space and action space. $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a transition function mapping state s and action a to the next state distribution $P(s'|s, a)$. $R_E : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the bounded extrinsic reward function. $\gamma \in [0, 1)$ is the discount factor. And $\mu \in \Delta(\mathcal{S})$ is the initial state distribution.



AP-MARL: fail to cause the victim to take poor actions

IMAP (ours): successfully block the victim and make it fall

Fig. 2: The victim (in blue) is attacked by an adversarial opponent (in red) in the multi-agent environment YouShallNotPass. The adversary is trained via (**top**) AP-MARL or (**bottom**) IMAP. AP-MARL learns to statically collapse on the ground and fail to block the victim. In contrast, our IMAP learns a stronger adversarial skill to intercept the victim.

b) Multi-Agent RL Tasks: For multi-agent tasks, we focus on two-player zero-sum competition games. A two-player zero-sum competition game can be formulated as a Markov Game $M = ((\mathcal{S}^{\nu}, \mathcal{S}^{\alpha}), (\mathcal{A}^{\nu}, \mathcal{A}^{\alpha}), P, (R_E, -R_E), \gamma, \mu)$. $\mathcal{S}$ and $\mathcal{A}$ stand for the state and action space repsectively. Here, we use α to represent the adversary and ν the victim. $P : \mathcal{S}^{\nu} \times \mathcal{S}^{\alpha} \times \mathcal{A}^{\nu} \times \mathcal{A}^{\alpha} \to \Delta(\mathcal{S}^{\nu}, \mathcal{S}^{\alpha})$ is the transition funtion where $\Delta(\mathcal{S}^{\nu}, \mathcal{S}^{\alpha})$ is the probability distribution space over $\mathcal{S}^{\nu}$ and $\mathcal{S}^{\alpha}$. $R_E : \mathcal{S}^{\nu} \times \mathcal{S}^{\alpha} \times \mathcal{A}^{\nu} \times \mathcal{A}^{\alpha} \times \mathcal{S}^{\nu} \times \mathcal{S}^{\alpha} \to \mathbb{R}$ is the bounded instant extrinsic reward function for the victim policy, and $-R_E$ is the corresponding extrinsic reward function for the adversarial agent according to the zero-sum assumption. $\gamma \in [0, 1)$ is the common discount factor determining the horizon of the Markov Game, and $\mu \in \Delta(\mathcal{S}^{\nu}, \mathcal{S}^{\alpha})$ is the initial state distribution.

c) Policy Optimization: We use Proximal Policy Optimization (PPO) [25] for AP learning. The objective function of PPO is defined as:

$$
\begin{aligned}
J^{\text{PPO}}(\pi) = \mathbb{E}_{s,a} \min \Bigg\{ & \frac{\pi(a|s)}{\pi_k(a|s)} \hat{A}, \\
& \text{clip}\left(\frac{\pi(a|s)}{\pi_k(a|s)}; 1 - \epsilon, 1 + \epsilon \right) \hat{A} \Bigg\},
\end{aligned} \tag{1}
$$

where 1) the density ratio $\frac{\pi(a|s)}{\pi_k(a|s)}$ is the importance weighting; 2) the clipping function $\text{clip}(x; 1 - \epsilon, 1 + \epsilon) = \begin{cases} 1 - \epsilon, & x \le 1 - \epsilon \\ 1 + \epsilon, & x \ge 1 - \epsilon \\ x, & \text{otherwise} \end{cases}$ is to make sure that the policy gradient is zero when $|1 - \frac{\pi(a|s)}{\pi_k(a|s)}| \ge \epsilon$; 3) the advantage function $\hat{A}$ is estimated by Generalized Advantage Estimation (GAE) [26] to reduce the variance of policy gradient estimation, that is, $\hat{A}(s_t) = \sum_{l=0}^{\infty} (\gamma\lambda)^l (R_E(s_t, a_t, s_{t+1}) + \gamma V^{\pi_k}(s_{t+l+1}) - V^{\pi_k}(s_{t+l}))$; and 4) the outer minimization operator ensures the objective function J^{PPO} is a lower bound of the objective $\mathbb{E}_{s,a} A$. Intuitively, this objective function makes sure the new and old policies are not so different. PPO then utilizes multiple steps of mini-batch Stochastic Gradient Descent (SGD) on J^{PPO} with a dataset $\mathcal{D} = \{(s, a, r_E, s')\}$ collected by the old policy π_k and use regression to update the value function V^{π}.

III. Threat Model

We adopt a black-box threat model for AP learning in both single- and multi-agent RL tasks. We describe the threat model from three aspects: objective, knowledge, and capabilities of the adversary.

A. Objective of the Adversary

In both single- and multi-agent RL tasks, the goal of the attacker is to learn an optimal AP π^{α} that can *minimize* the test-time expected episode rewards of the deployed black-box victim policy π^{ν}. We denote the adversarial state distribution induced by both π^{α} and π^{ν} as $d^{\pi^{\alpha}} = d^{\pi^{\alpha};\pi^{\nu}}$ to make the math notations concise since π^{ν} is held fixed. We define the test-time expected episode rewards of the victim policy as

$$
J_E^{\nu}(d^{\pi^{\alpha}}) = \sum_s d^{\pi^{\alpha}} \hat{r}_E^{\nu}, \tag{2}
$$

where $\hat{r}_E^{\nu}$ is the surrogate reward of the victim policy since we assume the adversary cannot access the training-time reward r_E^{ν} of the victim policy. r_E^{ν} may contain complex reward shaping terms, while $\hat{r}_E^{\nu}$ is a simple indicator that the victim completes the task (e.g., runs far enough in locomotion or reaches the target position in navigation and manipulation) in the single-agent environment or win the competitive game in the multi-agent environment, that is, $\hat{r}_E^{\nu} = \mathbb{1}(\text{the victim succeeds})$. The objective of the AP is then

$$
J^{\text{AP}}(\pi^{\alpha}) = -J_E^{\nu}(d^{\pi^{\alpha}}). \tag{3}
$$

B. Knowledge of the Adversary

In both single- and multi-agent RL tasks, the knowledge of the attacker is black-box, and the deployed victim policy network is assumed to be held fixed. Specifically, we assume that the adversary does not know the following information of the victim policy π^{ν}: 1) training-time hyperparameters; 2) *training-time rewards* r_E^{ν} and the *value function* $V^{\pi^{\nu}}$; 3) test-time model architecture, parameters and activations.

Clarification. Here, we clarify the assumptions made above. The first and third assumptions are typical black-box assumptions adopted by all existing back-box AP learning methods, including SA-RL, AP-MARL, and Wu et al.'s method. For the second assumption, it is worth noting that only the victim policy network is utilized during the deployment phase in RL tasks. Therefore, for evasion attacks against RL, it is reasonable that both the training-time rewards r_E^{ν} and the value function $V^{\pi^{\nu}}$—which are only used in the training phase—are unknown to the adversary. SA-RL relaxed the second assumption, resulting in a weaker threat model.

C. Capabilities of the Adversary

Here, we introduce the adversary's capabilities separately in single- and multi-agent tasks since their transition functions are different, as stated in Section II.

a) Single-Agent RL Tasks: The attacker can add small perturbations to the victim policy's inputs. We model the attacker as a state adversary $\pi^{\alpha}(\cdot|s)$, which can generate an adversarial perturbation $a^{\alpha} \sim \pi^{\alpha}$ based on the victim's current state s^{ν}. The perturbation a^{α} is bounded in an ℓ_p norm ball with a constant small radius ϵ, that is, $\|a^{\alpha}\|_p \leq \epsilon$. The transition function under this threat model becomes $P^{\alpha}(s^{\nu}_{t+1}|s^{\nu}_t, a^{\alpha}_t) = P(s^{\nu}_{t+1}|s^{\nu}_t, \pi^{\nu}(s^{\nu}_t + a^{\alpha}_t))$.

b) Multi-Agent RL Tasks: We focus on two-player zero-sum competitive games. The attacker can control an opponent agent α to battle with the victim agent ν, as visualized in Fig. 2. Since the victim policy is held fixed, the two-player Markov game M reduces to a single-player MDP $M^{\alpha} = ((\mathcal{S}^{\nu}, \mathcal{S}^{\alpha}), \mathcal{A}^{\alpha}, P^{\alpha}, (R_E, -R_E), \gamma, \mu)$. The transition function under this treat model becomes $P^{\alpha}(s^{\alpha}_{t+1}|s^{\alpha}_t, a^{\alpha}_t) = P(s^{\nu}_{t+1}, s^{\alpha}_{t+1}|s^{\nu}_t, s^{\alpha}_t, \pi^{\nu}(s^{\nu}_t, s^{\alpha}_t), a^{\alpha}_t)$. In each interaction step, the victim agent takes its action $\pi^{\nu}(s^{\nu}_t, s^{\alpha}_t)$ based on the current environment state $(s^{\nu}_t, s^{\alpha}_t)$, and the adversarial agent samples its action $a^{\alpha}_t \sim \pi^{\alpha}(\cdot|s^{\nu}_t, s^{\alpha}_t)$ simultaneously.

On the Adversary's Capabilities. In sum, the attacker can obtain the environment's current state—s^{ν} in single-agent tasks and $(s^{\nu}_t, s^{\alpha}_t)$ in multi-agent tasks—and maliciously influence the victim policy. In single-agent tasks, the adversary can directly inject perturbations a^{α}_t to the inputs of the victim policy, i.e., $\pi^{\nu}(s^{\nu}_t + a^{\alpha}_t)$; in multi-agent tasks, the adversary can indirectly influence the victim policy by generating adversarial observations s^{α}_t with an opponent agent, i.e., $\pi^{\nu}(s^{\nu}_t, s^{\alpha}_t)$.

IV. Proposed Attack

In this section, we introduce the detailed techniques of IMAP. We start with the design of its regularizer-based optimization objective and the resulting RL problem for regularizer-based black-box AP learning. We then introduce four types of principled and well-motivated adversarial intrinsic regularizers as particular design cases. Following this, we derive the details of how to solve the policy optimization problem of IMAP. Finally, we introduce a novel bias-reduction mechanism for IMAP to relieve the potential distraction caused by adversarial intrinsic regularizers.

A. Optimization Objective of IMAP

Under the black-box threat model, as discussed in Section III, maximizing the objective of the AP $J^{\text{AP}}(\pi^{\alpha})$ with trivial exploration methods like SA-RL and AP-MARL suffers from sample inefficiency and suboptimal solutions. To address these issues, we propose adversarial intrinsic regularizers, which intrinsically motivate the AP to explore novel states so as to uncover the potential vulnerabilities of the victim policy and learn stronger attacking skills. To make a trade-off between exploration (i.e., maximizing the adversarial intrinsic regularizer) and exploitation (i.e., maximizing the objective of the AP), we introduce a *regularization* approach by incorporating the adversarial intrinsic regularizer $J_I(d^{\pi^{\alpha}})$ into the objective of the AP $J^{\text{AP}}(\pi^{\alpha})$. The resulting optimization objective of IMAP is formulated as follows:

$$J^{\text{IMAP}}(\pi^{\alpha}) = J^{\text{AP}}(\pi^{\alpha}) + \tau_k J_I(d^{\pi^{\alpha}}), \tag{4}$$

where τ_k represents the temperature parameter that determines the strength of the regularization.

It is worth noting that our formulation for the optimization objective of IMAP $J^{\text{IMAP}}(\pi^{\alpha})$ is general. The adversarial intrinsic regularizer $J_I(d^{\pi^{\alpha}})$ can be a general function depending on the current adversarial state distribution $d^{\pi^{\alpha}}$ and all past adversarial state distribution $\{d^{\pi^{\alpha}_i}\}^{k}_{i=0}$. The adversarial intrinsic regularizer is designed to encourage the exploration of the AP in a principled manner. We present four types of adversarial intrinsic regularizers for IMAP as specific design cases in the following section.

Based on Eq. (4), the resulting policy optimization problem of IMAP becomes

$$\max J^{\text{IMAP}}(\pi^{\alpha}), \text{ s.t. } \pi^{\alpha} \in \arg\max J^{\text{AP}}(\pi^{\alpha}). \tag{5}$$

The constraint is necessary so as to ensure that, at convergence, the optimal AP for $J^{\text{IMAP}}(\pi^{\alpha})$ is optimal for the objective of the adversary $J^{\text{AP}}(\pi^{\alpha})$. We name this constraint the adversarial optimality constraint.

Uncovering Potential Vulnerabilities of the Victim Policy. Before delving into the design of adversarial intrinsic regularizers, it is crucial to define the potential vulnerabilities of a victim policy. Formally, what we are looking for is a state region in the victim policy's state space, that is, $\mathcal{W}^{\nu} \in \mathcal{S}^{\nu}$, where $\hat{r}^{\nu}_E$ is small or zero. In other words, $\mathcal{W}^{\nu}$ is the state region that all sub-optimal trajectories of the victim policy pass through. Thus, uncovering the potential vulnerabilities of the victim policy entails diverting the victim policy from its optimal trajectories. This definition is consistent with the objective of the AP. In single-agent tasks, since $\mathcal{S}^{\alpha} = \mathcal{S}^{\nu} \ni \mathcal{W}^{\nu}$ and $\|a^{\alpha}\|_p \leq \epsilon$, we can encourage the adversary to directly explore $\mathcal{S}^{\alpha}$ to find $\mathcal{W}^{\nu}$. On the contrary, in multi-agent tasks, $\mathcal{S}^{\nu} \neq \mathcal{S}^{\alpha}$, and the victim's and the adversary's states s^{ν} and s^{α} are coupled by the transition function $P^{\alpha}(s^{\alpha}_{t+1}|s^{\alpha}_t, a^{\alpha}_t)$ derived in Section III. Thus, we can design adversarial intrinsic regularizers in $\mathcal{S}^{\alpha}$, $\mathcal{S}^{\nu}$, or $(\mathcal{S}^{\alpha}, \mathcal{S}^{\nu})$, to encourage the AP to uncover $\mathcal{W}^{\nu}$.

B. Adversarial Intrinsic Regularizer Design

We now introduce how to design appropriate adversarial intrinsic regularizers for black-box AP learning. Recall the objective of the adversary is to maximize the objective of the AP $J^{\text{AP}}(\pi^{\alpha})$. Existing black-box AP learning methods in both single-agent and multi-agent RL tasks typically rely on the heuristic exploration technique, which involves random perturbation on the outputs of the AP without considering the learning process of the AP. However, these methods have been shown to be sample-inefficient and are prone to converging towards suboptimal solutions due to premature exploitation, particularly in sparse-reward tasks. To overcome these limitations, we design four types of adversarial intrinsic regularizers to stimulate the exploration of the AP, including SC-driven, PC-driven, R-driven, and D-driven regularizers.

1) State-Coverage-Driven Regularizer: The first type of adversarial intrinsic regularizer we design is the State-Coverage-driven (SC) regularizer. The SC-driven regularizer aims to encourage the AP to maximize the adversarial SC by maximizing the entropy of the adversarial state distribution d^{π^α}. For instance, in a single-agent navigation RL task, the AP learned via IMAP-SC can disrupt the victim policy by enticing it to move randomly in the whole map. The SC-driven regularizer for single-agent tasks can be defined as follows:

$$J_I^{\mathrm{SC}}(d^{\pi^\alpha}) = -\textstyle\sum_s d^{\pi^\alpha} \ln d^{\pi^\alpha}. \tag{6}$$

For multi-agent RL tasks, to uncover potential vulnerabilities $\mathcal{W}^\nu$ of the victim policy, we can 1) lure the victim policy into uniformly covering $\mathcal{S}^\nu$, and 2) encourage the adversary itself to uniformly cover $\mathcal{S}^\alpha$. To accomplish this, we define the marginal state distribution $d_{\mathcal{Z}}^{\pi}(z) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(\Pi_{\mathcal{Z}}(s_t) = z|\mu,\pi)$, where $\Pi_{\mathcal{Z}}$ is an operator mapping the full state into a projection space $\mathcal{Z}$. The SC-driven regularizer for multi-agent tasks is then formulated as

$$J_I^{\text{SC-M}}(d^{\pi^\alpha}) = (1-\xi)J_I^{\mathrm{SC}}(d_{\mathcal{S}^\alpha}^{\pi^\alpha}) + \xi J_I^{\mathrm{SC}}(d_{\mathcal{S}^\nu}^{\pi^\alpha}), \tag{7}$$

where ξ is a constant for balancing the two sub-objectives.

2) Policy-Coverage-Driven Regularizer: Next, we introduce the Policy-Coverage-driven (PC) adversarial intrinsic regularizer, which aims to intrinsically motivate the adversary to divert the victim policy from its past (optimal) trajectories so as to uncover its potential vulnerabilities efficiently. We define the adversarial explored regions, or adversarial PC, as the sum of all historical adversarial state distributions $\rho^\alpha = \sum_{i=1}^{k} d^{\pi_i^\alpha}$. For single-agent tasks, we design the PC-driven adversarial intrinsic regularizer as follows:

$$J_I^{\mathrm{PC}}(d^{\pi^\alpha}) = -\textstyle\sum_s \rho^\alpha \ln \rho^\alpha. \tag{8}$$

This regularizer can be regarded as the entropy of the adversarial PC. It encourages the adversary to visit novel regions where ρ^α is small.

With the definition of the marginal state distribution in the previous section, we can design the novel PC-driven regularizer for multi-agent tasks

$$J_I^{\text{PC-M}}(d^{\pi^\alpha}) = (1-\xi)J_I^{\mathrm{PC}}(d_{\mathcal{S}^\alpha}^{\pi^\alpha}) + \xi J_I^{\mathrm{PC}}(d_{\mathcal{S}^\nu}^{\pi^\alpha}). \tag{9}$$

Here, the first term encourages the adversary to visit novel states beyond the explored regions, while the second term aims to derail the victim from its optimal trajectories. The parameter ξ is used to balance the two sub-objectives.

3) Risk-Driven Regularizer: Besides the SC- and PC-driven adversarial intrinsic regularizers, we propose a novel Risk-driven (R) adversarial intrinsic regularizer for black-box AP learning. The concept of the risk is inspired by safety RL [20], where a cost function $c(s)$ is designed to constrain the behavior of the agent. For instance, when there exists a dangerous state s^{d} in the state space, the cost function can be designed as $c(s) = -\|s - s^{\mathrm{d}}\|$, penalizing the agent when it is close to s^{d}. By minimizing the expected cost function, the agent can be guided to stay away from s^{d}. In the context of evasion attacks, the attacker can maliciously select a potentially vulnerable state of the victim and lure the victim to approach this state. We refer to the state strategically selected by the adversary α for the victim ν as the adversarial state $s^{\nu(\alpha)} \in \mathcal{W}^\nu$. The corresponding cost function for the AP is then $c^\alpha(s) = -\|\Pi_{\mathcal{S}^\nu}(s) - s^{\nu(\alpha)}\|$. Here, we use the projector $\Pi_{\mathcal{S}^\nu}(s) \in \mathcal{S}^\nu$ to project the environment's full state s into the victim policy's state space $\mathcal{S}^\nu$ since R only concerns the victim's states. The R-driven adversarial intrinsic regularizer for both single- and multi-agent tasks is then

$$J_I^{\mathrm{R}}(d^{\pi^\alpha}) = -\textstyle\sum_s d^{\pi^\alpha} \|\Pi_{\mathcal{S}^\nu}(s) - s^{\nu;\alpha}\|. \tag{10}$$

Since all trajectories of the victim start from its initial state s_0^ν, we have $s_0^\nu \in \mathcal{W}^\nu$. Thus, a natural choice of $s^{\nu(\alpha)}$ is s_0^ν.

4) Divergence-Driven Regularizer: We now introduce the fourth type of adversarial intrinsic regularizer, the D-driven adversarial intrinsic regularizer. The design of the D-driven regularizer is based on policy diversity [21] and [22]. The objective of the D-driven regularizer is to intrinsically motivate the AP π^α to continuously deviate from its past policies $\{\pi_i^\alpha\}_{i=1}^k$, promoting diversity of the AP's behaviors and preventing the AP from being trapped in a local sub-optimal strategy. Note that we design the D-driven regularizer solely from the adversary's perspective, aiming to investigate whether this proprioceptive design can also help the AP discover potential vulnerabilities of the victim policy. Instead of randomly selecting an old policy from $\{\pi_i^\alpha\}_{i=1}^k$, we introduce one adversarial mimic policy $\pi^{\alpha;m}$ which has the same neural architecture as the AP π^α and imitates the behaviors of these past policies $\{\pi_i^\alpha\}_{i=1}^k$ by minimizing their average KL-divergence over all states, i.e., $\min\sum_s D_{\mathrm{KL}}(\pi^{\alpha;m}, \{\pi_i^\alpha\}_{i=1}^k)$. We then define the D-driven regularizer for both single- and multi-agent tasks as follows:

$$J_I^{\mathrm{D}}(d^{\pi^\alpha}) = \textstyle\sum_s d^{\pi^\alpha} D_{\mathrm{KL}}\left(\pi^\alpha, \pi^{\alpha;m}\right). \tag{11}$$

By maximizing $J_I^{\mathrm{D}}(d^{\pi^\alpha})$, the AP is encouraged to constantly deviate from its past policies to explore novel states in $\mathcal{S}^\nu$ to uncover $\mathcal{W}^\nu$ in a proprioceptive manner.

Relationships Between the Four Types of Adversarial Intrinsic Regularizers. Here, we clarify the relationships between the four types of adversarial intrinsic regularizers, i.e., SC-, PC-, R-, and D-driven adversarial intrinsic regularizers. They can be classified into two major categories, i.e., knowledge-based and data-based, depending on whether the regularizer involves only the agent's latest experiences (i.e., d^{π^α}) or the whole historical knowledge (i.e., $\{d^{\pi_i^\alpha}\}_{i=1}^k$ or $\{\pi_i^\alpha\}_{i=1}^k$). Thus, it is clear that SC- and R-driven regularizers belong to data-based since they only involve the adversary's latest state distribution d^{π^α}. In contrast, PC- and D-driven regularizers belong to knowledge-based because they both employ the adversary's all historical knowledge ρ^α or $\{\pi_i^\alpha\}_{i=1}^k$.

State Density Approximation. To solve the optimization problem of IMAP, it is crucial to approximate the adversarial state density d^{π^α} that all four regularizers we design involve. In the existing literature, there are two main types of methods for approximating state density, i.e., prediction-error-based and K-nearest-neighbour (KNN) estimation. Prediction-error-based methods, such as ICM [27] and RND [28], directly estimate the inverse of state density using the prediction errors of a neural network. However, these methods suffer from forgetting problems [29], [30]. We thus turn to the KNN method, a more efficient and stable nonparametric technique [31]. KNN estimates the state density via the inverse of the distance between a state and its K-nearest neighbor. Intuitively, the larger the distance, the smaller the state density (the sparser the samples). To estimate d^{π^α}, we cannot directly sample trajectories using the unsolved new policy π^α. In turn, we use the old policy π_k^α to sample trajectories since PPO guarantees that $D_{\text{KL}}(P^{\pi_k^\alpha} \| P^{\pi^\alpha}) \leq \delta$. Thus, the estimated adversarial SC is given by $d^{\pi^\alpha}(s) \approx 1/\|s - s^*_{\mathcal{D}_k}\|$. Here, $\mathcal{D}_k$ is a replay buffer containing trajectories sampled by only the latest old policy π_k^α, and $s^*_{\mathcal{D}} \in \mathcal{D}$ is the K-nearest state of s in $\mathcal{D}$. Similarly, the adversarial PC can be estimated via $\rho^\alpha(s) \approx 1/\|s - s^*_{\mathcal{B}}\|$, where $\mathcal{B} = \bigcup_{i=1}^{k} \mathcal{D}_i$ is the union replay buffer that contains all historical sampled trajectories. Note that one does not need to maintain the functional forms of all the old APs to estimate ρ^α. Instead, it is sufficient to sequentially store the trajectories sampled by the old policy π_i^α at the i-th iteration of the policy optimization into $\mathcal{B}$ and use the replay buffer $\mathcal{B}$ to estimate the policy cover ρ^α based on the KNN method.

C. Solving the IMAP Optimization Problem

We now present how to solve the policy optimization problem of IMAP defined in Eq. (5). It is easy to verify that $J^{\text{IMAP}}(\pi^\alpha)$ is a concave function of d^{π^α}. Thus, we can leverage the Frank-Wolfe algorithm to solve this problem. Specifically, it iteratively solves the following problem

$$\pi_{k+1}^\alpha \in \arg\max \left\langle d^{\pi^\alpha}, \nabla J^{\text{IMAP}}(\pi_k^\alpha) \right\rangle \tag{12}$$

to constructs a sequence of $\pi_0^\alpha, \pi_1^\alpha, \ldots$ that converges to an optimal AP $\pi^{\alpha *}$. The right-hand side of Eq. (12) is also known as the Frank-Wolfe gap [32]. Maximizing the Frank-Wolfe gap is equivalent to finding a policy π^α that maximizes the expected episode rewards, which is in proportion to $\nabla J^{\text{IMAP}}(\pi_k^\alpha)$. Hence, we can obtain the adversarial intrinsic bonus as follows:

$$r_I^\alpha = \nabla J_I(d^{\pi^\alpha}), \tag{13}$$

and can derive the objective of IMAP based on Eq. (1)

$$\begin{aligned} J^{\text{IMAP}}(\pi^\alpha) = \mathbb{E}_{s,a} \min \Bigg\{ & \frac{\pi^\alpha(a|s)}{\pi_k^\alpha(a|s)} \left(\hat{A}_E + \tau_k \hat{A}_I\right), \\ & \text{clip}\left(\frac{\pi^\alpha(a|s)}{\pi_k^\alpha(a|s)}; 1-\epsilon, 1+\epsilon\right) \left(\hat{A}_E + \tau_k \hat{A}_I\right) \Bigg\}, \end{aligned} \tag{14}$$

where $\hat{A}_E$ and $\hat{A}_I$ are the estimated extrinsic and intrinsic advantage functions.

D. Reducing Bias in IMAP

Though adversarial intrinsic regularizers can intrinsically motivate the AP to uncover the potential vulnerabilities of the victim policy, they may introduce bias to the optimal AP. In other words, the adversarial optimality constraint in Eq. (5) may not hold, i.e., $\arg\max J^{\text{IMAP}}(\pi^\alpha) \neq \arg\max J^{\text{AP}}(\pi^\alpha)$. One common practice to reduce this bias is to perform a hyperparameter search to find the best sequences of the temperature parameter $\{\tau_i\}_{i=0}^T$ for different tasks. However, an exhaustive hyperparameter search is computationally expensive and sample-intensive.

Algorithm 1 IMAP

```
Initialize the AP $\pi^\alpha$
Initialize replay buffers $\mathcal{B}$ and $\mathcal{D}$
Initialize counters $t = 0$ and $k = 0$
Initialize the temperature parameter $\tau_0 = 1$
Choose an adversarial intrinsic regularizer $J_I(d^{\pi^\alpha})$
while $t < T$ do
  # Sampling Stage
  Collect $\mathcal{D} = \{(s, a, -\hat{r}_E^\nu, s')\}$ using $\pi_k^\alpha$ against $\pi^\nu$
  Update the replay buffer $\mathcal{B} = \mathcal{B} \cup \mathcal{D}$
  Update the sample counter $t = t + \text{len}(\mathcal{D})$
  # Optimizing Stage
  Compute the intrinsic bonus $r_I^\alpha$ via Eq. (13)
  Estimate advantages $\hat{A}_E$ and $\hat{A}_I$ via GAE
  Update the AP $\pi^\alpha$ via Eq. (14)
  Update value functions $V_E^\alpha$ and $V_I^\alpha$ via regression
  if use BR then
    Update $\tau_k$ via Eq. (16) and Eq. (17)
  end if
  Update the iteration counter $k = k + 1$
end while
```

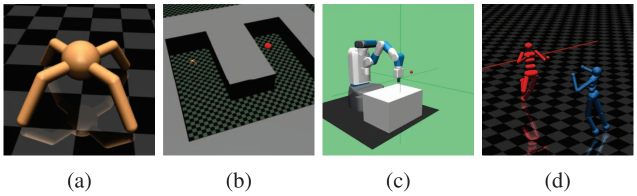
On Hyperparameter Search. Task-dependent temperature schedulers commonly utilize hyperparameter search to generate a sequence of the temperature parameter $\{\tau_i\}_{i=0}^T$ in advance, e.g., the exponentially decreasing scheduler $\tau_k = \beta(1-\rho)^k$ where both β and ρ are the hyperparameters to control the shape of the exponential function. Determining the optimal hyperparameters requires an expensive grid search. As the number of hyperparameters increases, the cost of the hyperparameter search grows exponentially. Conversely, our BR is a task-independent self-adaptive temperature scheduler that contains only one hyperparameter.

To address this challenge, we propose a novel adaptive Bias-Reduction (BR) method to ensure the adversarial optimality constraint. It is essential to balance the extrinsic objective and the intrinsic regularizer to ensure the maximization of the extrinsic objective (i.e., meeting the adversarial optimality constraint) rather than prioritizing the intrinsic regularizer at the end of the training process. Specifically, we propose an approximate adversarial optimality constraint, that is,

$$\begin{aligned} & \max J^{\text{AP}}(\pi^\alpha) + J_I(d^{\pi^\alpha}) \\ & \text{s.t. } J^{\text{AP}}(\pi^\alpha) >= J^{\text{AP}}(\pi_k^\alpha). \end{aligned} \tag{15}$$

Once the approximate adversarial optimality constraint is satisfied, we have $J^{\text{AP}}(\pi_{k+1}^\alpha) \geq J^{\text{AP}}(\pi_k^\alpha)$, that is, the objective of the AP J^{AP} monotonically increases.

To solve this soft-constrained optimization problem, we leverage the Lagrangian method to convert it into an unconstrained min-max optimization problem. The Lagrangian of Eq. (15) is $\mathcal{L}(\pi^\alpha, \lambda) = J^{\text{AP}}(\pi^\alpha) + J_I(d^{\pi^\alpha}) + \lambda(J^{\text{AP}}(\pi^\alpha) -$



(a) (b) (c) (d)

Fig. 3: Rendered pictures of typical MuJoCo environments. (a) the locomotion environment Ant; (b) the navigation environment AntUMaze where the red point is the goal position; (c) the manipulation environment FetchReach where the red point is the goal position; (d) the two-player zero-sum competitive game YouShallNotPass where the blue human is the victim and the red is the adversary.

$J^{\text{AP}}(\pi_k^{\alpha})) \propto J^{\text{AP}}(\pi^{\alpha}) + (1+\lambda)^{-1} J_I(d^{\pi^{\alpha}})$ where λ is the Lagrangian multiplier, and the corresponding dual problem is $\min_{\lambda \geq 0} \max_{\pi^{\alpha}} \mathcal{L}(\pi^{\alpha}, \lambda)$. By defining the temperature parameter τ_k as

$$\tau_k = (1+\lambda_k)^{-1}, \tag{16}$$

we have $J^{\text{IMAP}}(\pi^{\alpha}) = \mathcal{L}(\pi^{\alpha}, \lambda_k)$. We alternatively update π^{α} and λ, that is,

$$\begin{aligned} &\pi_{k+1}^{\alpha} \in \arg\max J^{\text{IMAP}}(\pi^{\alpha}) \\ &\lambda_{k+1} = \lambda_k - \eta(J^{\text{AP}}(\pi_{k+1}^{\alpha}) - J^{\text{AP}}(\pi_k^{\alpha})), \end{aligned} \tag{17}$$

to ensure that J^{IMAP} and J^{AP} are monotonically increased.

The form of the Lagrangian implies an interpretation for balancing the objective of the AP J^{AP} and the adversarial intrinsic regularizer J_I. At the beginning of training, $\lambda_0 = 0$ and $\tau_0 = 1$, the AP focuses on exploring novel states to discover the potential vulnerabilities of the victim policy $\mathcal{W}^{\nu}$ via maximizing the sum of the objective of the AP J^{AP} and the adversarial intrinsic regularizer J_I. When λ grows as the training progresses, the AP pays more attention to exploiting the uncovered states in $\mathcal{W}^{\nu}$ via directly maximizing J^{AP}.

V. Experiments

We conduct comprehensive experiments in various types of RL tasks to evaluate our IMAP's attacking capacity and generalization with four types of adversarial intrinsic regularizers and verify the effectiveness of our bias-reduction method.

A. Task Descriptions

In this section, we describe the details of the selected tasks. We evaluate our IMAP on both single- and multi-agent RL tasks. All environments are implemented based on the OpenAI Gym library and MuJoCo. For single-agent environments, we choose 1) four dense-reward locomotion tasks, including Hopper, Walker2d, HalfCheetah, and Ant [1], [8], [23], [24]; 2) six sparse-reward locomotion tasks, including SparseHopper, SpasreWalker2d, SparseHalfCheetah, SparseAnt, SparseHumaonidStandup, and SparseHumanoid [18], [33]; 3) two sparse-reward navigation tasks, AntUMaze and Ant4Room [34], [35]; and 4) one sparse-reward manipulation task, FetchReach [36]. We choose two challenging two-player zero-sum competitive games, YouShallNotPass and KickAndDefend [11], [15], [16], [37]–[39], as our multi-agent environments.

a) Criteria for Task Selection: The selection of tasks in our experiments is based on two main criteria. First, all tasks must be typical and have been adopted in former AP- and RL-related research works. This ensures that our evaluation is based on well-established benchmarks and allows for meaningful comparisons with existing methods. Second, the types of tasks must be diverse to evaluate the attack capacity and generalization of our IMAP comprehensively. In total, we selected 13 single-agent tasks and 2 multi-agent tasks that meet these criteria. Notably, tasks such as Ant, SpasreAnt, and YouShallNotPass have been used in multiple attacking and defense methods, making them suitable for comparative evaluations. The selected single-agent tasks cover three types: locomotion, navigation, and manipulation. We specifically include a manipulation task to demonstrate that our IMAP can efficiently learn optimal black-box APs to attack agents in tasks other than locomotion tasks. To further increase task diversity and evaluate IMAP's efficacy in multi-agent environments, we include two competitive games. These games involve victim agents with diverse skills, such as running and kicking. Moreover, the dimension of the environment state varies across the single-agent tasks, ranging from 11 (Hopper) to 378 (Humanoid). In the multi-agent tasks, the dimension of the environment state grows to 378x2. This variation in the state space dimension allows us to assess the performance of IMAP across tasks with different levels of complexity. Overall, our selected tasks ensure a comprehensive evaluation of IMAP.

b) Evaluation Metrics: In our evaluation of single-agent tasks, we use the average episode rewards of the victim policy under attacks as the primary evaluation metric. This is a common metric used to assess the performance of the victim policy. A lower average episode reward indicates a more successful evasion attack, as the victim policy is less effective in achieving its intended goals. For multi-agent tasks, we follow the previous works and report the attacking success rate of the AP. The attacking success rate is defined as $ASR = \frac{\text{\# of episodes where the adversary wins}}{\text{\# of total episodes}}$. It is easy to observe that $ASR = J^{\text{AP}} + 1$. A higher ASR indicates a stronger AP.

c) Single-Agent Tasks: In dense-reward single-agent tasks, the victim agent is expected to run as fast as possible and live as long as possible. According to the threat model in Section III, the adversary cannot access the victim's training-time reward r_E^{ν} which contains complex reward shaping terms like $-\omega_a^{\nu}\|a^{\nu}\|^2$ and $-\omega_f^{\nu}\|f^{\nu}\|^2$. Instead, the adversary uses the surrogate reward $\hat{r}_E^{\nu}$. In sparse-reward single-agent tasks, the victim agent is required to reach a certain goal at the end of the episode. In four locomotion tasks, the victim agent starts from the initial position and must move forward across a distant line to complete the task. The Ant environment is rendered in Fig. 3a. The episode is terminated once the victim agent gets the extrinsic reward or enters an unhealthy state. In two navigation tasks, the victim agent must navigate an Ant on different maps to reach a target region instead of always moving forward. This kind of task is thus known as more challenging than locomotion tasks like Ant and SparseAnt [34]. The environment AntUMaze is shown in Fig. 3b. In the manipulation task FetchReach, the robot arm is reset to an initial posture in each episode, and the victim agent is demanded to control the arm to move the end effector to a target position. FetchReach is visualized in Fig. 3c.

d) Multi-Agent Tasks: In YouShallNotPass, two humanoid robots are initialized facing each other. The victim policy controls the runner (in blue), while the AP controls the blocker (in red), as visualized in Fig. 3d. The victim wins if it reaches the finish line within 500 timesteps, whereas the adversary wins if the victim does not. KickAndDefend is a soccer penalty shootout between two humanoid robots. The victim policy controls the kicker (in blue), and the AP controls the goalie (in red). The victim wins if it shoots the ball into the red gate; otherwise, the adversary wins. The victim policies were trained via self-playing against random old versions of their opponents.

B. Baselines and Implementation

We now introduce the baselines used in our experiments.

a) Single-Agent Tasks: We select SA-RL [1], the state-of-the-art black-box AP learning method for single-agent tasks, as the baseline. The original SA-RL relaxes the black-box assumption and requires the training-time reward r_E^{ν} to learn the optimal AP. To ensure a fair comparison, we implement both SA-RL and IMAP with the same simple surrogate reward $-\hat{r}_E^{\nu}$ defined in Section III-A across all tasks. Moreover, all evasion attack methods for single-agent tasks in our experiments use the same attacking budget ϵ in each task. To justify the choice of the baseline, here we discuss other related AP methods for single-agent tasks. Yu et al.'s method [40] is tailored for video games. Sun et al.'s method [14] and Mo et al.'s method [17] fall under the category of white-box AP methods, as they necessitate access to the accurate model architecture and parameters of the victim policy. What is more, SA-RL outperforms MaxDiff and Robust Sarsa in their original paper [1]. Thus, SA-RL is the most suitable choice for our baseline in single-agent tasks.

b) Multi-Agent Tasks: We choose AP-MARL [11] as the baseline, which is recognized as the state-of-the-art black-box AP learning method for multi-agent tasks. To justify this choice, we mention here other existing AP methods for multi-agent tasks. As highlighted in Section III, our threat model is the same as that of AP-MARL. Wu et al.'s method [15], while adopting the same threat model, introduces the requirement of training an extra surrogate victim model. This added complexity, however, results in only a marginal improvement when compared to AP-MARL. As reported in their original paper, Wu et al.'s method achieves an ASR of only 60% in YouShallNotPass, while AP-MARL achieves an ASR of 59% in our experiments. Gong et al.'s method [38] demands access to the training-time value function $V^{\pi^{\nu}}$ of the victim policy, thereby violating our threat model. In addition, Gong et al.'s method reports an ASR of only 76% in YouShallNotPass. In contrast, our method, IMAP-PC+BR, achieves a substantially higher ASR of 83.91% in the same environment without any relaxation of the black-box assumptions. Guo et al.'s method [16] extends AP-MARL to non-zero-sum competitive games and is the same as AP-MARL in zero-sum competitive games. Thus, AP-MARL is the ideal baseline for two-player zero-sum competitive games, which are our primary focus.

C. Evaluation Results

In this section, We report our main results. At a high level, our experiments reveal the following set of observations:

IMAP vs. SA-RL: IMAP dominates SA-RL against most (15 out of 22) models and is comparable in the reset across all dense-reward single-agent tasks. Among all types of IMAP attacks, IMAP-PC achieves the best average performance.

Generalization: IMAP excels in terms of generalization, surpassing SA-RL across diverse types of tasks, including locomotion, navigation, and manipulation tasks.

Choice of Adversarial Intrinsic Regularizers: IMAP-PC is a suitable choice for a novel task since it exhibits superior generalization across our proposed four types of IMAP attack.

Effect of BR in IMAP: The use of the balancing method BR in IMAP proves effective in enhancing the attacking performance, particularly when the adversarial intrinsic bonuses strongly distract the adversary.

IMAP vs. AP-MARL: IMAP-PC+BR significantly outperforms AP-MARL in two zero-sum competitive games.

Hyperparameter Sensitivity: IMAP displays resilience to variations in two newly introduced hyperparameters within reasonable bounds, i.e., ξ in Eq. (9) and η in Eq. (17).

Evading Defense Methods: IMAP successfully evades two different types of robust training defense methods, namely, adversarial training and robust regularizer.

1) Performance in Dense-Reward Tasks: We first discuss the results of IMAP v.s. SA-RL in dense-reward tasks shown in Table I.

a) IMAP Outperforms SA-RL: As shown in Table I, IMAP performs the best against most (15 out of 22) models (bolded results in each row) and is comparable to SA-RL in the rest. Here are some points that need to be explained. Firstly, when attacking the vanilla PPO models, IMAP significantly outperforms SA-RL in Walker (895 vs. 1253) and Ant (188 vs. 351) and performs equally in Hopper (both 80) and HalfCheetah (both 0). This underscores that when the victim policy has evident vulnerabilities, both SA-RL, utilizing the ad-hoc dithering exploration method, and IMAP, employing principled adversarial intrinsic regularizers, can readily identify and exploit these vulnerabilities to disrupt the victim. However, when there are more subtle vulnerabilities that elude trivial exploration methods, IMAP remains capable of efficiently discovering such vulnerabilities and further compromising the performance of the victim policy. Secondly, it is reasonable that there is no big difference between the performance of IMAP and SA-RL against certain models (e.g., 7 comparable cases beyond the 15 of 22 outperforming cases), such as 4377 vs. 4376 against Walker RADIAL and 4202 vs.

TABLE I: Average episode rewards J_E^ν ± standard deviation of one vanilla model trained via PPO and five robust models trained using various defense methods over 300 episodes in four dense-reward locomotion tasks under no attack, random attack, SA-RL, and our four types of IMAP attacks. We **bold** the best attack result (the lowest value) in each row and also report the average Attack Performance across all models. IMAP—the best of the four types of IMAP attacks—outperforms SA-RL against most (15 out of 22) models and exhibits similar performance in the reset across all dense-reward single-agent tasks. Among all attacks, IMAP-PC performs the best regarding the average performance.

Env.	Victim	No Attack	Random	SA-RL	IMAP-SC	IMAP-PC	IMAP-R	IMAP-D
Hopper 11D 0.075	PPO (va.)	3167 ± 542	2101 ± 793	$\mathbf{80 \pm 2}$	$\mathbf{80 \pm 2}$	$\mathbf{80 \pm 2}$	$\mathbf{80 \pm 2}$	$\mathbf{80 \pm 2}$
	ATLA	2559 ± 958	2153 ± 882	875 ± 145	689 ± 132	$\mathbf{639 \pm 48}$	672 ± 120	808 ± 170
	SA	3705 ± 2	2710 ± 801	1826 ± 897	$\mathbf{1282 \pm 68}$	1346 ± 85	1714 ± 1176	2278 ± 1144
	ATLA-SA	3291 ± 600	3165 ± 576	1585 ± 469	1685 ± 512	$\mathbf{1536 \pm 392}$	1807 ± 642	1823 ± 527
	RADIAL	3740 ± 44	3729 ± 100	$\mathbf{1622 \pm 408}$	2194 ± 672	1647 ± 398	1871 ± 498	1895 ± 551
	WocaR	3616 ± 99	3633 ± 30	1850 ± 530	2140 ± 612	$\mathbf{1646 \pm 337}$	2917 ± 495	1832 ± 493
Average Across Victims		3346	2915	1306	1345	**1149**	1510	1452
Walker 17D 0.05	PPO (va.)	4472 ± 635	3007 ± 1200	1253 ± 468	1002 ± 391	$\mathbf{895 \pm 450}$	2966 ± 956	947 ± 160
	ATLA	3138 ± 1061	3384 ± 1056	1163 ± 464	1035 ± 614	$\mathbf{991 \pm 500}$	1599 ± 742	1385 ± 590
	SA	4487 ± 61	4465 ± 39	3927 ± 162	4196 ± 231	$\mathbf{3072 \pm 1304}$	4083 ± 155	3820 ± 39
	ATLA-SA	3842 ± 475	3927 ± 368	3508 ± 66	3144 ± 995	$\mathbf{2868 \pm 1145}$	3620 ± 143	3469+650
	RADIAL	5251 ± 12	5184 ± 42	$\mathbf{4376 \pm 1229}$	4562 ± 941	4377 ± 1147	4584 ± 1021	4474 ± 1187
	WocaR	4156 ± 495	4244 ± 157	2871 ± 1153	3178 ± 1168	2874 ± 1085	$\mathbf{2740 \pm 1162}$	2859 ± 1078
Average Across Victims		4224	4035	2850	2853	**2513**	3265	2826
HalfCheetah 17D 0.15	PPO (va.)	7117 ± 98	5486 ± 1378	$\mathbf{0 \pm 0}$	$\mathbf{0 \pm 0}$	$\mathbf{0 \pm 0}$	56 ± 147	$\mathbf{0 \pm 0}$
	ATLA	5417 ± 49	5388 ± 34	$\mathbf{1696 \pm 1352}$	2451 ± 1352	1711 ± 1357	1996 ± 965	1765 ± 1357
	SA	3632 ± 20	3619 ± 18	2997 ± 22	2996 ± 24	$\mathbf{2984 \pm 20}$	3390 ± 62	3000 ± 27
	ATLA-SA	6157 ± 852	6164 ± 603	$\mathbf{4170 \pm 664}$	4311 ± 412	4202 ± 726	4395 ± 728	4231 ± 681
	RADIAL	4724 ± 14	4731 ± 42	1654 ± 1312	1669 ± 1326	$\mathbf{1641 \pm 1298}$	1791 ± 1278	2563 ± 1496
	WocaR	6032 ± 68	5969 ± 149	4257 ± 1254	$\mathbf{3734 \pm 1512}$	4026 ± 1374	4782 ± 105	4759 ± 487
Average Across Victims		5513	5226	2462	2433	**2427**	2730	2720
Ant 111D 0.15	PPO (va.)	5687 ± 758	5261 ± 1005	351 ± 110	310 ± 184	212 ± 244	$\mathbf{188 \pm 135}$	284 ± 195
	ATLA	4894 ± 123	4541 ± 691	$\mathbf{0 \pm 0}$	428 ± 63	70 ± 128	696 ± 24	$\mathbf{0 \pm 0}$
	SA	4292 ± 384	4986 ± 452	2698 ± 822	2720 ± 879	$\mathbf{2643 \pm 851}$	2722 ± 994	2746 ± 831
	ATLA-SA	5359 ± 153	5366 ± 104	3125 ± 207	3228 ± 190	3156 ± 302	$\mathbf{2611 \pm 213}$	3125 ± 182
Average Across Victims		5058	5039	1544	1672	**1520**	1554	1539

4170 against HalfCheetah ATLA-SA. This can be attributed to three factors: 1) the victim agent's strong robustness, making its vulnerabilities difficult to detect even with adversarial intrinsic regularizers; 2) the potential distraction introduced by the adversarial intrinsic regularizers, which may divert the adversarial policy from maximizing its core objective; 3) the worst cases in these 7 comparable tasks are easier to uncover compared to the other 15 outperforming tasks, causing that both IMAP and SA-RL can successfully discover the worst cases in these tasks and exhibit similar performances. Note that the distraction phenomenon is more apparent in sparse-reward tasks. We delve deeper into it in the following section.

b) Choice of Adversarial Intrinsic Regularizers: Black-box robustness evaluation of the RL agent is a trial-and-error process where the agent knows nothing about the black-box victim model, including the training method, as stated in Section III. We thus recommend starting the evaluation of a new black-box victim agent with IMAP-PC as the first trial since the experiments show it behaves well on average. As evident from Table I, IMAP-PC demonstrates the best average performance among all types of IMAP attacks. It notably reduces the average performance of all victim models by 65.66%, 40.52%, 55.97%, and 69.94% in Hopper, Walker, HalfCheetah, and Ant, respectively. For a comprehensive assessment of the robustness of a black-box victim policy, it is reasonable to explore multiple types of IMAP attacks. An essential insight from the results in Table I is that the type of potential vulnerabilities of the victim policy is not tied to the training method of that policy. For instance, the vulnerabilities of the ATLA-SA model in Ant can be identified via IMAP-R (reducing performance from 5359 to 2611), while the vulnerabilities of the ATLA-SA model in Walker can be exposed through IMAP-PC (reducing performance from 3842 to 2868). This pattern holds for other victim policy training methods as well. Therefore, it is advisable to try all adversarial intrinsic regularizers to discover potential vulnerabilities of the victim policy thoroughly. Additionally, we do not recommend combining multiple adversarial intrinsic regularizers since they may violate each other and make the adversary struggle.

c) On the Large Standard Deviation in the Performance of AP Attacks: The presence of substantial variance in the performance of reinforcement learning algorithms is a well-acknowledged phenomenon. This variability primarily stems from the inherent variance in policy gradient estimation [25], [41]. Given that the objective of the AP is maximized by PPO, it is unsurprising that the results exhibit large standard

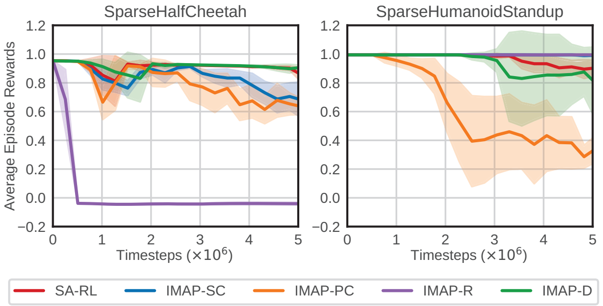


Fig. 4: Curves of test-time attacking results of SA-RL and four types of IMAP attacks on six sparse-reward locomotion tasks. IMAP-R significantly outperforms SA-RL in SparseHopper and SpareWalker2d; IMAP-PC significantly surpasses SA-RL in SparseHalfCheetah and SparseHumanoidStandup.

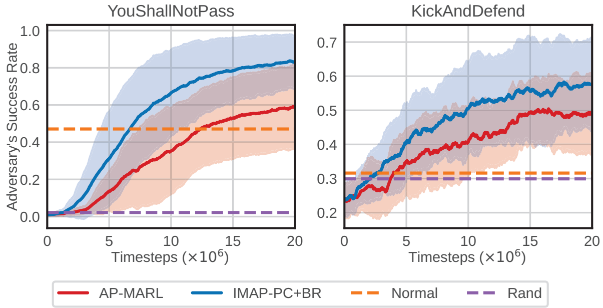


Fig. 5: Learning curves of AP-MARL and IMAP-PC+BR in two-player zero-sum competitive games. IMAP-PC+BR outperforms AP-MARL by a large margin.

deviations. It is noteworthy that this phenomenon of significant standard deviation is not exclusive to IMAP but has also been reported in the original papers of SA-RL [1] and AP-MARL [11]. Importantly, variances do not significantly affect the application of AP methods. In practice, attackers have the flexibility to train multiple APs using various seeds and select the best one to attack the victim.

2) Performance in Sparse-Reward Tasks.: We now discuss the results in spares-reward single-agent tasks shown in Fig. 4 and Table II.

a) Attacking Capacity and Generalization: The results presented in Table II underscore IMAP's superior performance, outperforming SA-RL across all sparse-reward tasks. Additionally, as shown in Fig. 4, IMAP exhibits a significant advantage over SA-RL. In particular, SA-RL struggles to learn any effective attacking strategy with the trivial exploration method in SparseWalker2d. In contrast, IMAP-R efficiently discovers an optimal AP, leading to a remarkable reduction in the victim's average episode rewards, from 0.95 to -0.04, using only 0.5M samples (10× less than the 5M training sample budget). In SparseHumanoidStandup, SA-RL costs 5M samples to decrease the victim's performance from 0.99 to 0.88, while our IMAP-PC decreases the victim's performance to 0.4 within 2.5M samples (2× less). In terms of generalization, IMAP consistently diminishes the performance of victim agents across all three types of tasks, including locomotion, navigation, and manipulation tasks. Moreover, IMAP surpasses SA-RL in terms of the average performance across tasks (in the last line of Table II). These findings highlight the superior attacking capacity and generalization of IMAP compared to the baseline SA-RL.

b) Choice of Adversarial Intrinsic Regularizers: Again, we discuss the choice of the adversarial intrinsic regularizers in sparse-reward tasks. From Table II, we observe that IMAP-PC mainly excels in locomotion and manipulation tasks; IMAP-D performs the best in navigation tasks; and IMAP-R stands out in partial locomotion tasks. These findings lead us to conclude that the suitability of an adversarial intrinsic regularizer is closely tied to the type of task. It is expected that different types of victim agents possess distinct potential vulnerabilities. For instance, in locomotion tasks like SparseHopper and SparseWalker, where the victim policy is dynamically unstable and prone to fall into unhealthy states under perturbations, the R-driven adversarial intrinsic regularizer is more likely to reveal these vulnerabilities. Considering that the average performance of IMAP-PC is the best, one may try IMAP-PC first and then other types of IMAP.

c) Effect of Bias-Reduction: The results presented in Table II reveal that bias-reduction (BR) yields notable performance improvements for IMAP in several sparse-reward tasks. Specifically, in Ant4Rooms, IMAP-R is heavily distracted by the R-driven regularizers. With the incorporation of BR, IMAP-R's performance is substantially enhanced, elevating it from 0.74 to 0.22. Note that 0.22±0.48 (R) indicates that this result is achieved by IMAP-R+BR. Similarly, IMAP-PC benefits from BR, improving its performance from 0.37 to 0.22 and emerging as the top-performing attack in AntUMaze. These outcomes underscore the efficacy of BR in augmenting the performance of IMAP in sparse-reward tasks.

3) Performance in Competitive Games: In this section, we discuss the results of IMAP v.s. AP-MARL in multi-agent tasks, as shown in Fig. 5.

a) IMAP-PC+BR Outperforms AP-MARL: Building upon the insights gained from single-agent tasks, we delve into the performance of IMAP-PC+BR in two-player zero-sum competitive games in comparison to AP-MARL. Remarkably, IMAP-PC+BR consistently outperforms AP-MARL by a substantial margin. As illustrated in Fig. 5, IMAP-PC+BR consistently surpasses AP-MARL, substantially elevating the ASR from 59.64% to an impressive 83.91%. This remarkable enhancement can be attributed to the acquisition of more natural attacking behavior in YouShallNotPass, as evidenced in Fig. 2. In KickAndDefend, the game imposes constraints on the adversary (the goalie), confining it to a square region before the gate. Even within these constraints, IMAP manages to enhance the ASR from 47.02% to 56.96%. These results

TABLE II: Average episode rewards J_E^{ν} ± standard deviation of the victim policies over 1000 episodes across nine sparse-reward tasks, including six locomotion tasks (starting with 'S.'), two navigation tasks AntUMaze and Ant4Rooms, and one manipulation task, under nine attacks, including one baseline attack SA-RL, four types of IMAP attacks, and four types of IMAP+BR attacks. We **bold** the best attack performance under each row. IMAP dominates SA-RL across all nine tasks (highlighted by ▒). BR improves the attack performance of IMAP further in (4 out of 9) tasks.

Env.	No Attack	Random	SA-RL	IMAP-SC	IMAP-PC	IMAP-R	IMAP-D	IMAP+BR
S.Hopper	0.95 ± 0.00	0.95 ± 0.00	0.01 ± 0.32	0.00 ± 0.30	0.16 ± 0.45	-0.03 ± 0.00	-0.02 ± 0.28	**-0.05 ± 0.22** (PC)
S.Walker	0.95 ± 0.00	0.94 ± 0.11	0.85 ± 0.23	0.66 ± 0.44	0.63 ± 0.45	**-0.04 ± 0.01**	0.91 ± 0.06	0.80 ± 0.32 (R)
S.HalfCheetah	0.98 ± 0.00	0.98 ± 0.00	0.30 ± 0.51	0.17 ± 0.45	**0.04 ± 0.35**	0.98 ± 0.00	0.33 ± 0.51	0.06 ± 0.37 (SC)
S.Ant	0.99 ± 0.00	0.98 ± 0.10	0.12 ± 0.42	0.23 ± 0.48	0.27 ± 0.49	0.43 ± 0.49	0.12 ± 0.42	**0.10 ± 0.40** (D)
S.HumanStand	0.99 ± 0.00	0.99 ± 0.00	0.88 ± 0.32	0.99 ± 0.05	**0.23 ± 0.50**	0.99 ± 0.00	0.80 ± 0.42	0.36 ± 0.54 (PC)
S.Humanoid	0.96 ± 0.00	0.93 ± 0.21	0.49 ± 0.50	0.46 ± 0.50	0.40 ± 0.49	**0.24 ± 0.44**	0.45 ± 0.5	0.35 ± 0.48 (PC)
AntUMaze	0.98 ± 0.00	0.98 ± 0.00	0.32 ± 0.52	0.30 ± 0.51	0.37 ± 0.52	0.97 ± 0.10	0.28 ± 0.51	**0.19 ± 0.47** (PC)
Ant4Rooms	0.91 ± 0.23	0.91 ± 0.00	0.34 ± 0.51	0.32 ± 0.51	0.40 ± 0.52	0.74 ± 0.43	0.24 ± 0.48	**0.22 ± 0.48** (R)
FetchReach	0.99 ± 0.00	0.99 ± 0.00	0.31 ± 0.50	-0.10 ± 0.00	**-0.10 ± 0.00**	0.73 ± 0.42	0.51 ± 0.49	**-0.10 ± 0.00** (PC)
Average	0.97	0.96	0.40	0.34	0.28	0.56	0.40	**0.21**

reinforce the superior efficacy of IMAP-PC+BR in multi-agent tasks compared to AP-MARL, highlighting the effectiveness of the PC-driven regularizer in uncovering potential vulnerabilities in the victim policy.

b) Fundamental Reasons for Outperforming AP-MARL: The primary distinction lies in their exploration strategies employed during the training stage. AP-MARL utilizes a heuristic dithering exploration strategy, while IMAP+PC is intrinsically motivated by the PC-driven regularizer. The PC-driven regularizer allows IMAP to uncover the vulnerabilities of the victim $\mathcal{W}^{\nu}$ more efficiently through a larger coverage on the victim and the adversary's joint state space $(\mathcal{S}^{\nu}, \mathcal{S}^{\alpha})$.

c) Ablation Study on Hyperparameters: We conducted an in-depth investigation into the impact of IMAP's two newly introduced hyperparameters: the updating step size η of the Lagrangian multiplier in Eq. (17) and the constant ξ for balancing the two sub-objectives in Eq. (9). Fig. 6 and Fig. 7 reveal the performance of IMAP-PC+BR under different hyperparameter settings in single- and multi-agent tasks separately. Fig. 6 demonstrates that IMAP is insensitive to η when $\eta \in \{1, 5, 10, 50\}$. A larger updating step size leads to better performance within this range. Fig. 7 shows that IMAP is also robust to changes of $\xi \in \{0.5, 1\}$. Recall that $J_I^{\text{SC-M}}(d^{\pi^{\alpha}}) = (1-\xi) J_I^{\text{SC}}(d^{\pi^{\alpha}}_{\mathcal{S}^{\alpha}}) + \xi J_I^{\text{SC}}(d^{\pi^{\alpha}}_{\mathcal{S}^{\nu}})$. Fig. 7 indicates that $J_I^{\text{SC}}(d^{\pi^{\alpha}}_{\mathcal{S}^{\alpha}})$ is critical for the performance of IMAP-PC. Note that when these hyperparameters go beyond rational ranges (i.e., [1,50] for the updating step size η and (0,1] for the balancing constant ξ), the performance of IMAP may significantly deteriorate. For instance, when $\xi = 0$, the ASR of IMAP drops from the optimum 83.91% to the baseline ASR of 60%.

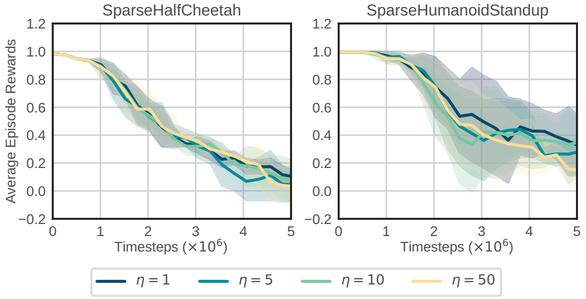


Fig. 6: Ablation study on the hyperparameter η of IMAP.

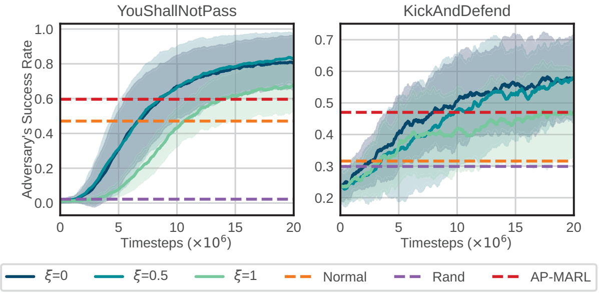


Fig. 7: Ablation study on the hyperparameter ξ of IMAP.

VI. Defense Methods Against IMAP

In this section, we explore potential defense methods against IMAP and evaluate IMAP's effectiveness against two main types of defense methods.

Possible Defense Methods Against IMAP. There are four categories of methods for RL agents to defend against evasion attacks: adversarial training, robust regularizer, randomized smoothing, and active detection. Adversarial training in the context of RL closely resembles its counterpart in DNN. It involves optimizing the policy under either gradient-based evasion attacks or the optimal AP. The adversary can have various access rights in the environment to robustify the victim agent against different types of uncertainties, e.g., directly injecting perturbations to the state or action or reward [1], [14], [42]–[45], adding disturbance forces or torques [12], or even changing the layout or dynamic property of the environment [46]. Robust regularizer aims to enhance the smoothness of the learned policy by upper-bounding the divergence of the action distributions under state perturbations [8], [23], [24], [47]. Randomized smoothing has been applied to analyze the robustness of reinforcement learning from a probabilistic perspective [9], [48]–[50]. Active detection strategies focus on identifying malicious samples by comparing the KL-divergence of the nominal action distribution and the predicted one [51] or using explainable AI techniques to identify critical time steps contributing to the victim agent's performance [52].

Evaluating IMAP Against Defense Methods. There are two types of defense methods based on the above analysis, i.e., robust training (adversarial training and robust regularizer) and test-time defense mechanisms (randomized smoothing and active detection). We focus on the first type of defense method against IMAP and leave the second type of defense method for future work. What is more, randomized smoothing and active detection may sacrifice the victim's test-time performance since they operate on the original inputs of the deployed victim policy. Robust regularizer methods include 1) SA [8], which improves the robustness of the victim agent via a smooth policy regularization (denoted as SA-regularizer for concision) on the victim policy solved by the convex relaxation technique; 2) RADIAL [23], which leverages an adversarial loss function based on bounds of the victim policy under bounded l_∞ attacks; and 3) WocaR [24], which directly estimates and optimizes the worst-case episode rewards also based on bounds of the victim policy under bounded l_∞ attacks. Two adversarial training methods include: 1) ATLA [1], which alternately trains the victim agent and an RL attacker with independent value and policy networks; and 2) ATLA-SA [1], which combines the training procedure of ATLA with the SA-regularizer and uses LSTM as the policy network. The results in Table I demonstrate our IMAP is effective in evading robust models trained by either adversarial training methods or robust regularizer methods. All victim models we adopt are publicly released. We report the average performance over 300 episodes to make the results statistically reliable. Notably, even against the state-of-the-art robust WocaR models, our IMAP can efficiently uncover their potential vulnerabilities via proper adversarial intrinsic regularizers under the black-box threat model, reducing their performance by 54.58%, 34.07%, and 38.10% in Hopper, Walker, and HalfCheetah respectively.

VII. Discussion

In this section, we provide an in-depth discussion of the sample efficiency of IMAP and identify the specific reinforcement learning engines or models that can benefit from the proposal of the IMAP.

On the Sample Efficiency. There are three key insights on the sample efficiency of IMAP. Firstly, the adversarial intrinsic regularizers (i.e., SC, PC, R, D) contribute more to the sample efficiency of IMAP compared to BR. Intuitively, when the potential vulnerabilities of the victim policy are extremely difficult to discover, it becomes challenging to learn an optimal adversarial policy with an inappropriate or no intrinsic regulator. Secondly, there is a trade-off between sample efficiency and performance. As shown in Table II, satisfactory results can be achieved by using the adversarial intrinsic regularizer PC alone, without the need for BR. Therefore, unless ultimate performance is sought, it is not necessary to increase the number of samples by 8x. Thirdly, IMAP-PC is based on policy cover theory that enjoys polynomial sample complexity [19]. Intuitively, it is aware of the agent's entire historical knowledge and explicitly deviates the victim policy from its optimal trajectories. Hence, IMAP-PC is more likely to discover the victim's worst cases than SA-RL which explores randomly.

RL Agents Benefiting From IMAP. There are various real-world scenarios for RL agents, e.g., Large Language Models (LLM) [53], autonomous driving [4], traffic control [54], industrial automation and manufacturing [55], dynamic treatment regimes [56], [57], and recommendation systems [58], [59]. IMAP is promising to evaluate these deployed black-box real-world RL engines or models. Here, we provide two appropriate cases. Firstly, to evaluate the robustness of a real-world victim autonomous driving RL agent, we can use IMAP to either generate stealthy sensor noise to disrupt the victim car [8] or control another malicious car to intercept the victim car to make a traffic jam or even accident [60]. Secondly, we can formulate the red-teaming tasks for LLM as a two-player competitive game, regarding the target LLM as the victim agent and the red-teaming language model as the adversarial policy [61]. In such a way, IMAP holds the potential for learning a strong, intrinsically motivated red-teaming adversarial policy to evaluate the robustness of the real-world commercial black-box LLM, e.g., GPT-4.

VIII. Conclusion

In this paper, we proposed a new regularizer-based AP learning method called IMAP to evaluate the robustness of test-time RL agents in single- and multi-agent environments under the black-box threat model. We presented four types of adversarial intrinsic regularizers that encourage the AP to explore novel states so as to uncover the potential vulnerabilities of the victim policy. We also introduced a novel balancing method, BR, to boost IMAP further. We conducted extensive evaluation experiments of IMAP across various types of tasks. The experimental results demonstrated that IMAP outperformed existing methods, including SA-RL and AP-MARL, in terms of attacking capability and generalization. We also empirically showed that BR effectively boosted IMAP in both single- and multi-agent environments. Moreover, we demonstrated that IMAP successfully evaded state-of-the-art defense methods, including adversarial training and robust regularizer methods. Additionally, our ablation study showed IMAP was insensitive to its main hyperparameters. Note that though our proposed four adversarial intrinsic regularizers covered the main branches of intrinsic motivation, one can still design new adversarial intrinsic regularizers for IMAP as needed. We leave this as future work.

Acknowledgment

We thank the anonymous reviewers and our shepherd, Dr. Stjepan Picek, for their helpful and valuable feedback, and Tencent Yunding Laboratory for providing computing resources and generous technical support. This work was partially supported by HK RGC under Grants (CityU 11218322, R6021-20F, R1012-21, RFS2122-1S04, C2004-21G, C1029-22G, and N_CityU139/21), the National Natural Science Foundation of China (U21B2018, 62161160337, 61822309, U20B2049, 61773310, U1736205, 61802166, 62276067), and Shaanxi Province Key Industry Innovation Program (2021ZDLGY01-02).

References

[1] H. Zhang, H. Chen, D. S. Boning, and C. Hsieh, "Robust reinforcement learning on state observations with learned optimal adversary," in *Proc. of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[2] Y. Huang and S. Wang, "Adversarial manipulation of reinforcement learning policies in autonomous agents," in *Proc. of International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2018.

[3] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 23, no. 2, pp. 740–759, 2020.

[4] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 23, no. 6, pp. 4909–4926, 2021.

[5] P. Buddareddygari, T. Zhang, Y. Yang, and Y. Ren, "Targeted attack on deep rl-based autonomous driving with learned visual patterns," in *Proc. of the 39th International Conference on Robotics and Automation (ICRA)*, pp. 10571–10577, 2022.

[6] B. Lütjens, M. Everett, and J. P. How, "Certified adversarial robustness for deep reinforcement learning," in *Proc. of the 3rd Annual Conference on Robot Learning (CoRL)*, pp. 1328–1337, 2020.

[7] M. Everett, "Neural network verification in control," in *Proc. of the 60th IEEE Conference on Decision and Control (CDC)*, pp. 6326–6340, 2021.

[8] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. S. Boning, and C. Hsieh, "Robust deep reinforcement learning against adversarial perturbations on state observations," in *Proc. of the 34th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 21024–21037, 2020.

[9] F. Wu, L. Li, Z. Huang, Y. Vorobeychik, D. Zhao, and B. Li, "CROP: Certifying robust policies for reinforcement learning through functional smoothing," in *Proc. of the 10th International Conference on Learning Representations (ICLR)*, 2022.

[10] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun, "Tactics of adversarial attack on deep reinforcement learning agents," in *Proc. of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3756–3762, 2017.

[11] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," in *Proc. of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[12] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *Proc. of the 34th International Conference on Machine Learning (ICML)*, pp. 2817–2826, 2017.

[13] J. Sun, T. Zhang, X. Xie, L. Ma, Y. Zheng, K. Chen, and Y. Liu, "Stealthy and efficient adversarial attacks against deep reinforcement learning," in *Proc. of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 5883–5891, 2020.

[14] Y. Sun, R. Zheng, Y. Liang, and F. Huang, "Who is the strongest enemy? towards optimal and efficient evasion attacks in deep RL," in *Proc. of the 10th International Conference on Learning Representations (ICLR)*, 2022.

[15] X. Wu, W. Guo, H. Wei, and X. Xing, "Adversarial policy training against deep reinforcement learning," in *Proc. of the 30th USENIX Security Symposium (USENIX Security)*, pp. 1883–1900, 2021.

[16] W. Guo, X. Wu, S. Huang, and X. Xing, "Adversarial policy learning in two-player competitive games," in *Proc. of the 38th International Conference on Machine Learning (ICML)*, pp. 3910–3919, 2021.

[17] K. Mo, W. Tang, J. Li, and X. Yuan, "Attacking deep reinforcement learning with decoupled adversarial policy," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 20, no. 1, pp. 758–768, 2022.

[18] E. Hazan, S. Kakade, K. Singh, and A. Van Soest, "Provably efficient maximum entropy exploration," in *Proc. of the 36th International Conference on Machine Learning (ICML)*, pp. 2681–2691, 2019.

[19] A. Agarwal, M. Henaff, S. M. Kakade, and W. Sun, "PC-PG: policy cover directed exploration for provable policy gradient learning," in *Proc. of the 34th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 13399–13412, 2020.

[20] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," in *Proc. of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[21] Z. Hong, T. Shann, S. Su, Y. Chang, T. Fu, and C. Lee, "Diversity-driven exploration strategy for deep reinforcement learning," in *Proc. of the 32nd Advances in Neural Information Processing Systems (NeurIPS)*, pp. 10510–10521, 2018.

[22] Y. Flet-Berliac, J. Ferret, O. Pietquin, P. Preux, and M. Geist, "Adversarially guided actor-critic," in *Proc. of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[23] T. P. Oikarinen, W. Zhang, A. Megretski, L. Daniel, and T. Weng, "Robust deep reinforcement learning through adversarial loss," in *Proc. of the 35th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 26156–26167, 2021.

[24] Y. Liang, Y. Sun, R. Zheng, and F. Huang, "Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning," in *Proc. of the 36th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 22547–22561, 2022.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[26] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[27] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proc. of the 34th International Conference on Machine Learning (ICML)*, pp. 2778–2787, 2017.

[28] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," in *Proc. of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[29] T. Zhang, H. Xu, X. Wang, Y. Wu, K. Keutzer, J. E. Gonzalez, and Y. Tian, "Noveld: A simple yet effective exploration criterion," in *Proc. of the 35th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 25217–25230, 2021.

[30] T. Zhang, P. Rashidinejad, J. Jiao, Y. Tian, J. E. Gonzalez, and S. Russell, "MADE: exploration via maximizing deviation from explored regions," in *Proc. of the 35th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9663–9680, 2021.

[31] H. Liu and P. Abbeel, "Behavior from the void: Unsupervised active pre-training," in *Proc. of the 35th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 18459–18473, 2021.

[32] M. Frank, P. Wolfe, *et al.*, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[33] M. Mutti, L. Pratissoli, and M. Restelli, "Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate," in *Proc. of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 9028–9036, 2021.

[34] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.

[35] B. Eysenbach, T. Zhang, S. Levine, and R. Salakhutdinov, "Contrastive learning as goal-conditioned reinforcement learning," in *Proc. of the 36th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 35603–35620, 2022.

[36] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, *et al.*, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," *arXiv preprint arXiv:1802.09464*, 2018.

[37] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch, “Emergent complexity via multi-agent competition,” in *Proc. of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[38] C. Gong, Z. Yang, Y. Bai, J. Shi, A. Sinha, B. Xu, D. Lo, X. Hou, and G. Fan, “Curiosity-driven and victim-aware adversarial policies,” in *Proc. of the 38th Annual Computer Security Applications Conference (ACSAC)*, pp. 186–200, 2022.

[39] S. Li, J. Guo, J. Xiu, P. Feng, X. Yu, A. Liu, W. Wu, and X. Liu, “Attacking cooperative multi-agent reinforcement learning by adversarial minority influence,” *arXiv preprint arXiv:2302.03322*, 2023.

[40] M. Yu and S. Sun, “Natural black-box adversarial examples against deep reinforcement learning,” in *Proc. of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 8936–8944, 2022.

[41] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, *et al.*, “Soft actor-critic algorithms and applications,” *arXiv preprint arXiv:1812.05905*, 2018.

[42] V. Behzadan and A. Munir, “Whatever does not kill deep reinforcement learning, makes it stronger,” *arXiv preprint arXiv:1712.09344*, 2017.

[43] E. Vinitsky, Y. Du, K. Parvate, K. Jang, P. Abbeel, and A. Bayen, “Robust reinforcement learning using adversarial populations,” *arXiv preprint arXiv:2008.01825*, 2020.

[44] K. L. Tan, Y. Esfandiari, X. Y. Lee, S. Sarkar, *et al.*, “Robustifying reinforcement learning agents via action space adversarial training,” in *Proc. of American Control Conference (ACC)*, pp. 3959–3964, 2020.

[45] J. Wu and Y. Vorobeychik, “Robust deep reinforcement learning through bootstrapped opportunistic curriculum,” in *Proc. of the 39th International Conference on Machine Learning (ICML)*, pp. 24177–24211, 2022.

[46] T. Chen, W. Niu, Y. Xiang, X. Bai, J. Liu, Z. Han, and G. Li, “Gradient band-based adversarial training for generalized attack immunity of a3c path finding,” *arXiv preprint arXiv:1807.06752*, 2018.

[47] Q. Shen, Y. Li, H. Jiang, Z. Wang, and T. Zhao, “Deep reinforcement learning with robust and smooth policy,” in *Proc. of the 37th International Conference on Machine Learning (ICML)*, pp. 8707–8718, 2020.

[48] B. G. Anderson and S. Sojoudi, “Certified robustness via locally biased randomized smoothing,” in *Proc. of the 4th Learning for Dynamics and Control Conference*, pp. 207–220, 2022.

[49] A. Kumar, A. Levine, and S. Feizi, “Policy smoothing for provably robust reinforcement learning,” in *Proc. of the 10th International Conference on Learning Representations (ICLR)*, 2022.

[50] M. Lutter, S. Mannor, J. Peters, D. Fox, and A. Garg, “Robust value iteration for continuous control tasks,” *arXiv preprint arXiv:2105.12189*, 2021.

[51] Y.-C. Lin, M.-Y. Liu, M. Sun, and J.-B. Huang, “Detecting adversarial attacks on neural network policies with visual foresight,” *arXiv preprint arXiv:1710.00814*, 2017.

[52] W. Guo, X. Wu, U. Khan, and X. Xing, “EDGE: explaining deep reinforcement learning policies,” in *Proc. of the 35th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12222–12236, 2021.

[53] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language models are few-shot learners,” in *Proc. of the 34th Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1877–1901, 2020.

[54] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, “Flow: Architecture and benchmarking for reinforcement learning in traffic control,” *arXiv preprint arXiv:1710.05465*, vol. 10, 2017.

[55] H. Oliff, Y. Liu, M. Kumar, M. Williams, and M. Ryan, “Reinforcement learning for facilitating human-robot-interaction in manufacturing,” *Journal of Manufacturing Systems*, vol. 56, pp. 326–340, 2020.

[56] J. Zhang and E. Bareinboim, “Near-optimal reinforcement learning in dynamic treatment regimes,” in *Proc. of the 33rd Advances in Neural Information Processing Systems (NeurIPS)*, pp. 13401–13411, 2019.

[57] J. Zhang, “Designing optimal dynamic treatment regimes: A causal reinforcement learning approach,” in *Proc. of the 37th International Conference on Machine Learning (ICML)*, pp. 11012–11022, 2020.

[58] L. Zou, L. Xia, Z. Ding, J. Song, W. Liu, and D. Yin, “Reinforcement learning to optimize long-term user engagement in recommender systems,” in *Proc. of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 2810–2818, 2019.

[59] M. M. Afsar, T. Crump, and B. Far, “Reinforcement learning based recommender systems: A survey,” *ACM Computing Surveys*, vol. 55, no. 7, pp. 1–38, 2022.

[60] A. Sharif and D. Marijan, “Adversarial deep reinforcement learning for improving the robustness of multi-agent autonomous driving policies,” in *Proc. of the 29th Asia-Pacific Software Engineering Conference*, pp. 61–70, 2022.

[61] Z.-W. Hong, I. Shenfeld, T.-H. Wang, Y.-S. Chuang, A. Pareja, J. R. Glass, A. Srivastava, and P. Agrawal, “Curiosity-driven red-teaming for large language models,” in *Proc. of the 12th International Conference on Learning Representations (ICLR)*, 2024.